What is claimed:

1. A method of programmatically calculating paths from a spatially-enabled database, comprising steps of:

identifying an origin and a destination;

determining a first street on which the origin is located and a second street on which the destination is location; and

computing a path from the origin on the first street to the destination on the second street using intersection data represented by street geometry data stored in the spatially-enabled database.

2. The method according to Claim 1, wherein the computing step further comprises iteratively performing, until completing the path, steps of:

computing a bounding box between the origin and the destination;

computing a shortest linear path ("SLP") between the origin and the destination; and

selecting an intersection point closest to the SLP to replace the origin for subsequent iterations of the iteratively performed steps,

wherein the path is complete when the street on which the origin is located intersects the street on which the destination is located.

3. The method according to Claim 2, wherein the selecting step gives preference to intersection points located within the bounding box.

4. The method according to Claim 2, wherein the intersection data is stored in a spatially-enabled table of the spatially-enabled database.

5. The method according to Claim 2, wherein the selecting step gives preference to intersection points whose SLP is not longer than a quantified percentage more than the SLP between the origin on the first street and the destination on the second street.

6. The method according to Claim 5, wherein the quantified percentage is a value specified by a user.

7. The method according to Claim 5, wherein the quantified percentage is a value obtained from a configuration file.

8. The method according to Claim 2, wherein the selecting step gives preference to intersection points whose bounding box is no more than a quantified percentage larger than the bounding box between the origin on the first street and the destination on the second street.

9. The method according to Claim 8, wherein the quantified percentage is a value specified by a user.

10. The method according to Claim 8, wherein the quantified percentage is a value obtained from a configuration file.

11. A system for programmatically calculating paths from a spatially-enabled database, comprising:

means for identifying an origin and a destination;

means for determining a first street on which the origin is located and a second street on which the destination is location; and

means for computing a path from the origin on the first street to the destination on the second street using intersection data stored in a spatially-enabled table of the spatially-enabled database.

12. The system according to Claim 12, wherein the means for computing further comprises means for iteratively performing, until completing the path, operation of:

means for computing a bounding box between the origin and the destination;

means for computing a shortest linear path ("SLP") between the origin and the destination; and

means for selecting an intersection point closest to the SLP to replace the origin for subsequent iterations of the iteratively performed operations,

wherein the path is complete when the street on which the origin is located intersects the street on which the destination is located.

13. A computer program product for programmatically calculating paths from a spatially-enabled database, the computer program product embodied on one or more computer-readable

media and comprising:

computer-readable program code means for identifying an origin and a destination;

computer-readable program code means for determining a first street on which the origin is located and a second street on which the destination is location; and

computer-readable program code means for computing a path from the origin on the first street to the destination on the second street using intersection data stored in a spatially-enabled table of the spatially-enabled database.

14. The system according to Claim 13, wherein the computer-readable program code means for computing further comprises computer-readable program code means for iteratively performing, until completing the path, operation of:

computer-readable program code means for computing a bounding box between the origin and the destination;

computer-readable program code means for computing a shortest linear path ("SLP") between the origin and the destination; and

computer-readable program code means for selecting an intersection point closest to the SLP to replace the origin for subsequent iterations of the iteratively performed operations,

wherein the path is complete when the street on which the origin is located intersects the street on which the destination is located.